Central Scientific
Computing Complex
Document Z-1a

MACSYMA USAGE AT LANGLEY

BY

SANDRA J. DELOATCH

JANUARY 1987

Prepared under Contract NAS1-16921

MACSYMA USAGE AT LANGLEY

Table of Contents

List of Tables

1.0 INTRODUCTION

MACSYMA is a large and versatile computer programing system written in LISP which is specifically designed for performing symbolic as well as numerical manipulations. MACSYMA's numerous capabilities for symbolic processing of mathematical expressions include: two-dimensional display of expressions, differentiation, integration, polynomial factorization, solutions of algebraic equations, manipulation of trigonometric functions, matrix and tensor manipulation, Taylor series expansion, and Laplace transforms.

MACSYMA was originally developed at Massachusetts Institute of Technology and is implemented on a DEC KL10 computer running under the ITS operating system in Cambridge, Massachusetts. Symbolics, Inc. of Massachusetts has converted MACSYMA to run on a number of other computers. Hampton University has obtained a license to use a version referred to as UNIX MACSYMA which runs on a DEC VAX 11/700 class computer under the UNIX[1] operating system. Arrangements have been made for LaRC affiliated researchers to access and use the Hampton University VAX 11/780 computer for symbolic mathematical processing. The system is generally available at all times.

This document will describe the use of MACSYMA at NASA, Langley Research Center. It will provide an introduction to the UNIX MACSYMA computer environment which features

- Information that new users need to begin using the system
- Resources available for new and experienced users
- Instructions and examples on selected features of MACSYMA

The following notational conventions will be used throughout this document:

<CR> Represents the return (carriage return) key.
<ESC> or <ALT> Represents the escape or altmode key.

^ Any character preceded by a caret (^) represents a control character. For example, ^Y is control Y. That is, hold down the control key while simultaneously pressing Y.

. Vertical ellipsis means that not all data the system would
. display is shown.
.

[1]UNIX is a trademark of Bell Laboratories

2.0 ACCESSING THE SYSTEM

In order to use MACSYMA you must obtain an account on the Hampton University computer. To apply for an account, complete a copy of the form entitled "REQUEST FOR SYMBOLIC MATHEMATICS COMPUTER ACCOUNT AT HAMPTON UNIVERSITY UNDER LaRC SPONSORSHIP" which is available from the Computer Applications Branch (CAB of ACD, Telephone (804) 865-3544). A copy of the form is also provided at the back of this document. Send the completed form to CAB. Once the application has been approved by CAB and Hampton University, the account will be established and the form will be returned to you.

Account holders on the Hampton University Computer may access MACSYMA through the Central Scientific Computing Complex dial-out facility. The following steps illustrate the log-in procedure.

1. Set terminal Baud rate to 1200

2. Press <CR> to signal Central Data Switch.

   Central Data Switch responds with

   ```
   NASA LANGLEY CENTRAL COMPUTERS
   ENTER RESOURCE
   ```

3. Type ADIAL <CR>

   When the connection to the dial-out facility has been made, "GO" appears on the screen. This message is followed by:

   ```
   DIAL-OUT FACILITY
   ENTER RESOURCE CODE:
   ```

   Proceed to step 4 if the above messages appear. If there is a problem in establishing a connection with the dial-out facility, one of the following messages may appear:

   ```
   UNAVAILABLE
   NO ANSWER
   UNAUTHORIZED
   WRONG SPEED
   NOT ASSIGNED
   BUSY, WAIT? Y/N    (Type Y <CR> to wait for a DIAL-OUT line or N
      <CR>, otherwise.)
   ```

4. Type HI <CR>

   The dial-out facility should respond with:

   DIALING

   then a slash (/) should appear on the screen. This indicates that the connection with Hampton University has been established. When the telephone line is busy or inoperative, one of the following messages will appear:

```
NUMBER BUSY
NUMBER INOPERATIVE
```

In such instances wait a few moments and try again from step 2.

5. Log-in

Once a connection has been made, communication is established with the UNIX operating system. At this point type <CR> and the system will prompt you for a log-in name (account designation) and password.

After entering your user name and password, the system responds with

```
Last login:  . . .
4.2 BSD UNIX #2
      .
      .
      .
```

and displays the prompting character, a percent sign (%)

If this is the first time you have used the system, you should immediately assign a password to your account. In order to designate a password, type

```
passwd <CR>
```

The system will prompt you for a password, and for a verification of the password. Passwords generally contain a minimum of 6 characters.

6. Load MACSYMA

To load MACSYMA type

```
MACSYMA <CR>
```

After a few seconds, MACSYMA will respond with

```
THIS IS UNIX MACSYMA RELEASE 309.2
(C) 1976, 1984 MASSACHUSETTS INSTITUTE OF TECHNOLOGY.
      .
      .
      .
(C1)
```

The label (C1) is automatically assigned to your first command. At this point expressions may be entered for evaluation by the system.

7. Log-out

   In order to log-out, type

   QUIT(); <CR>

   to exit MACSYMA. Then type

   logout <CR>

If your terminal is not connected to MICOM, you may use an acoustic coupler or modem at 300 or 1200 Baud to dial the telephone number (804-727-5743) of the Hampton University computer directly. When you hear the tone, place the telephone receiver in the acoustic coupler headset, or press the data button on the data set and hang up the receiver. At this point, you may proceed with steps 5 through 7 of the procedure described above.

3.0 SOME MACSYMA CAPABILITIES AND ILLUSTRATIONS OF SYNTAX

MACSYMA has been referred to as a mathematical consultant, and indeed its capabilities are unparalleled for many scientific and engineering applications. In addition, the system is fairly easy to use. After entering the necessary commands to get connected to the system, users are presented with command-lines (C-lines). At this point, FORTRAN-like expressions can be entered. Expressions may be composed of operators, variables, constants, or conditional statements.

The system uses the standard mathematical operators (+, -, *, /, ↑) and operator precedence. MACSYMA also features two non-standard operators. They are or ! for factorial and . for non-commutative multiplication. The only limitation on selecting variable names to be used in MACSYMA expressions is that they must not begin with a digit. Constants, too, are similar to those appearing in other programing languages. However, there are three constants which are used exclusively in MACSYMA. They are %P1, %E, and %I which represent π , e (base of natural logarithms), and $\sqrt{-1}$, respectively.

Values can be assigned to variables by using the assignment operator ":". Functions are defined by using the ":=" operator, or alternatively by the command define

```
define (function (arguments), body)
```

Once a syntactically correct expression has been entered, it will be evaluated immediately by the system. When the expression is terminated with ";", the system will display the result using a D-line number. If the expression is terminated with "$" (dollar-sign), the display is suppressed; however, the result is still associated with a D-line number. The D-line number represents a symbolic expression which may be manipulated or used in other expressions.

In addition to the usual interactive mode of operation, MACSYMA also provides a facility for executing commands stored in a file. The process requires that you first create a file containing the MACSYMA commands to be evaluated. Next, enter MACSYMA and type

```
batch ("file-specification"); <CR>
```

This facility is very useful for producing error-free listings of the solutions to complex or lengthy problems, since modifications can be made by using the text editor. Some other useful MACSYMA file commands are described in section 3.6.

3.1 Simplification Functions

Algebraic simplification is one of the most valuable capabilities of a computer algebra system. MACSYMA has numerous commands which can change the form or representation of an algebraic expression. Certain commands, however, are more appropriate for a specific class of functions. Table 3-1 contains a listing of selected functions and the simplification commands appropriate to each.

Function	Appropriate Simplification Commands
Polymonial	EXPAND, FACTOR, FACTORSUM
Rational	RATSIMP, FACTORSUM, RATEXPAND, XTHRU
Exponential/Logarithmic	RADCAN
Trigonometric	TRIGEXPAND, TRIGREDUCE

Table 3-1

The simplification commands in Table 3-1 require a single parameter as an argument. This parameter represents the expression or function to be simplified. Users may also create simplification rules by using the TELLSIMP or RATSUBST commands. An illustration of the use of selected simplification commands appears below.

Simplification Examples

Comments

```
(C1) p:(x-y)*(x^2 +x*y + y^2);
```

(* ":" is the MACSYMA assignment symbol *)

$$(D1) \qquad (X - Y)\ (Y^2 + X\,Y + X^2)$$

```
(C2) expand(p);
```

$$(D2) \qquad X^3 - Y^3$$

```
(C3) factor(p);
```

$$(D3) \qquad -(Y - X)\ (Y^2 + X\,Y + X^2)$$

```
(C4) q:x^2 - y^2;
```

$$(D4) \qquad X^2 - Y^2$$

```
(C5) expand(p*q);
```

$$(D5) \qquad Y^5 - X^2\,Y^3 - X^3\,Y^2 + X^5$$

```
(C6) factor(%);
```

(* "%" refers to the previous expression *)

$$(D6) \qquad (Y - X)^2\ (Y + X)\ (Y^2 + X\,Y + X^2)$$

```
(C7) r:x^2 + 3*y/(x^2 + 4*z);
```

$$(D7) \qquad \frac{3\,Y}{4\,Z + X^2} + X^2$$

```
(C8) s:(x^4-3*y-4*z-16*z^4)/(4*y^2*z+x^2*y^2);
```

$$(D8) \qquad \frac{-16\,Z^4 - 4\,Z - 3\,Y + X^4}{4\,Y^2\,Z + X^2\,Y^2}$$

```
(C9) t:r+s;
```

$$(D9) \qquad \frac{-16\,Z^4 - 4\,Z - 3\,Y + X^4}{4\,Y^2\,Z + X^2\,Y^2} + \frac{3\,Y}{4\,Z + X^2} + X^2$$

(C10) ratsimp(t);

(D10) $$- \frac{16 Z^{4} + (4 - 4 X^{2} Y^{2}) Z - 3 Y^{3} - X^{4} Y^{2} + 3 Y - X^{4}}{4 Y^{2} Z + X^{2} Y^{2}}$$

(C11) factorsum(%);

(D11) $$- \frac{4 Z (4 Z^{3} - X^{2} Y^{2} + 1) - Y (Y (3 Y + X^{4}) - 3) - X^{4}}{Y^{2} (4 Z + X^{2})}$$

(C12) u:r *s;

(D12) $$\frac{(- 16 Z^{4} - 4 Z - 3 Y + X^{4}) \left(\frac{3 Y}{4 Z + X^{2}} + X^{2}\right)}{4 Y^{2} Z + X^{2} Y^{2}}$$

(C13) xthru(%);

(D13) $$\frac{(X^{2} (4 Z + X^{2}) + 3 Y) (- 16 Z^{4} - 4 Z - 3 Y + X^{4})}{(4 Z + X^{2}) (4 Y^{2} Z + X^{2} Y^{2})}$$

(C14) w:exp(x^2)*log(x)*log(x^2+2*x+1)/(exp(x^2-y^2)*log((x+1)^2));

(D14) $$\frac{\mathrm{LOG}(X)\ \mathrm{LOG}(X^{2} + 2 X + 1)\ \%E^{Y^{2}}}{2\ \mathrm{LOG}(X + 1)}$$

(C15) radcan(%);

(D15) $$\mathrm{LOG}(X)\ \%E^{Y^{2}}$$

```
(C17) sin(h+g);
(D17)                                SIN(H + G)

(C18) trigexpand(%);

(D18)                    COS(G) SIN(H) + SIN(G) COS(H)

(C19) tr:sin((z-1)/3)/cos((z-1)/3)^2;
                                      Z - 1
                                  SIN(-----)
                                        3
(D19)                             ----------
                                     2 Z - 1
                                  COS (-----)
                                        3

(C20) trigreduce(tr);

                                   Z - 1      Z - 1
(D20)                         SEC(-----) TAN(-----)
                                   3          3

(C21) tellsimp(cos(x)^2,1-sin(x)^2);

(D21)                          [^RULE1, SIMPEXPT]

(C22) sin(x)/cos(x)^2;
                                    SIN(X)
(D22)                               ----------
                                         2
                                    1 - SIN (X)

(C23) trigreduce(%);

                                    2 SIN(X)
(D23)                               ----------
                                    COS(2 X) + 1

(C24) sec(x)^2 - tan(x)^2;
                                   2          2
(D24)                         SEC (X) - TAN (X)

(C25) ratsubst(1+tan(x)^2,sec(x)^2,%);
(D25)                                 1
```

3.2 Integration

The MACSYMA integration routines will find an indefinite integral for a large class of functions, if one exists. This class includes rational functions and functions that can be built upon from the rationals using exponential, logarithmic, trigonometric, and inverse trigonometric operations. There are three distinct stages involved in MACSYMA's evaluation of an indefinite integral. During the first stage, a test is made to determine if the integrand is of the from

$$f(g(x)) \cdot g'(x).$$

This is accomplished by checking to see if the derivative of some subexpression divides the integral. The method of solution, if the integrand is found to be of this form, is to search an integral table for an entry corresponding to f and substitute $g(x)$ for x in the integral of f. If the first stage fails, an attempt is made to match the integrand to a form for which a specific method (for example, trigonometric substitution) can be used. Failure at the second stage dictates that the general Risch decision procedure be used.

The INTEGRATE command is used to find the definite, or indefinite integral of an expression. The command requires a minimum of two parameters. The first parameter is the expression to be integrated and the second represents the variable of integration. For definite integration, two additional parameters are required. These parameters represent the lower and upper limits of integration. Illustrations of the INTEGRATE command appear below. (Note: 'INTEGRATE is the noun form of the command, and it causes the MACSYMA version integral symbol to be displayed. The EV command is used to evaluate the noun form of the command. More specifically, when

```
EV(%,INTEGRATE)
```

follows 'INTEGRATE, it means evaluate the previous expression with respect to integration.)

```
(C3) 'INTEGRATE(SEC(T)^2/(1+SEC(T)^2-3*TAN(T)),T);
                                  /          2
                                  [       SEC (T)
(D3)                              I ------------------------- dT
                                  ]                2
                                  / - 3 TAN(T) + SEC (T) + 1

(C4) EV(%,INTEGRATE);

(D4)                     LOG(TAN(T) - 2) - LOG(TAN(T) - 1)

(C5) 'INTEGRATE('INTEGRATE((X^2+1)/(x+z),x),z);
                                        / /  2
                                        [ [ X  + 1
(D5)                                    I I ------ dX dZ
                                        ] ] Z + X
                                        / /

(C6) RATSIMP(EV(%,INTEGRATE));
          3              3                          3        2         2
    (6 Z  + 18 Z + 6 X  + 18 X) LOG(Z + X) - 2 Z  - 6 X Z  + (3 X  - 18) Z
(D6) ----------------------------------------------------------------------
                                      18

(C7) 'INTEGRATE(%E^(2*X)/(1+%E^X),X);
                                          /    2 X
                                          [  %E
(D7)                                      I -------- dX
                                          ]   X
                                          / %E  + 1

(C8) EV(%,INTEGRATE);
                                       X          X
(D8)                                 %E  - LOG(%E  + 1)

(C9) 'INTEGRATE(X*LOG(X),X,0,1);
                                         1
                                        /
                                        [
(D9)                                    I  X LOG(X) dX
                                        ]
                                        /
                                         0

(C10) EV(%,INTEGRATE);

                                              1
(D10)                                       - -
                                              4
```

```
(C11) 'INTEGRATE(SIN(X)*COS(X),X,0,%PI/6);
                                    %PI
                                    ---
                                     6
                                   /
                                   [
(D11)                              I    COS(X) SIN(X) dX
                                   ]
                                   /
                                    0

(C12) EV(%,INTEGRATE);
                                             1
(D12)                                        -
                                             8

(C13) 'INTEGRATE((X+1)/SQRT(2*X-X^2),X);
                                    /
                                    [       X + 1
(D13)                               I  -------------- dX
                                    ]                2
                                    /  SQRT(2 X - X )

(C14) EV(%,INTEGRATE);


                                            2          2 - 2 X
(D14)                   - SQRT(2 X - X ) - 2 ASIN(-------)
                                                          2

(C15) 'INTEGRATE(X^4/(1-X^2)^(5/2),X);
                                     /       4
                                     [      X
(D15)                                I  ----------- dX
                                     ]        2 5/2
                                     /  (1 - X )

(C16) ratsimp(EV(%,INTEGRATE));
                               2        2                          3
                    SQRT(1 - X ) (3 X  - 3) ASIN(X) - 4 X  + 3 X
(D16)               ---------------------------------------------
                                          2       2
                               SQRT(1 - X ) (3 X  - 3)

(C17) 'INTEGRATE('INTEGRATE('INTEGRATE((z*y^2+1/(X-1)^2),X),y),z);
                         / / /
                         [ [ [    2            1
(D17)                    I I I (Y  Z + --------) dX dY dZ
                         ] ] ]                2
                         / / /          (X - 1)

(C18) EV(%,INTEGRATE);
                                    3  2
                                 X Y  Z     Y Z
(D18)                            -------  - -----
                                    6       X - 1

(D19)                                BATCH DONE
```

```
(C20) batch("[sjd]vcomp.max");

(C21) ASSUME(A)>0, B)>0, C)>0, E)>0, V)>0)$

(C22) J[1]: 1/((V^2+E^2)*(A*V^2+2*B*V+C)^(1/2))$

(C23) 'INTEGRATE(J[1],V);
                      /
                      [                1
(D23)                 I ------------------------------- dV
                      ]   2    2          2
                      / (V  + E ) SQRT(A V  + 2 B V + C)

(C24) EV(%,INTEGRATE);

Is  E  zero or nonzero?

nonzero;
                 2
Is  A E  - C  positive, negative, or zero?

pos;
                 2                  2
Is  (A E  - 2 B E - C) (A E  + 2 B E - C)  positive, negative, or zero?

    pos;
                 2
Is  A C - B  positive, negative, or zero?

neg;
```

$$
\begin{aligned}
(D24)\ & -\%I\ \mathrm{ASIN}\left(\frac{2\,\%I\,A\,E\,V}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)}\right) \\
& + \frac{2\,B\,V}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)} + \frac{2\,\%I\,B\,E}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)} \\
& + \frac{2\,C}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)}\Big) / (2\,E\,\mathrm{SQRT}(A\,E^2 - 2\,\%I\,B\,E - C)) \\
& - \%I\,\mathrm{ASIN}\left(\frac{2\,\%I\,A\,E\,V}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)} - \frac{2\,B\,V}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)}\right. \\
& \left. + \frac{2\,\%I\,B\,E}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)} - \frac{2\,C}{\mathrm{SQRT}(B^2 - A\,C)\,\mathrm{SQRT}(4\,V^2 + 4\,E^2)}\right) \\
& / (2\,E\,\mathrm{SQRT}(A\,E^2 + 2\,\%I\,B\,E - C))
\end{aligned}
$$

```
(C25) RADCAN(%);
                         2
(D25) (%I SQRT(A E  + 2 %I B E - C)

          2                                                 2     2
     SQRT(B  - A C) ((%I A E + B) V + %I B E + C) SQRT(V  + E )
ASIN(------------------------------------------------------------)
                              2    2            2    2
                     (A C - B ) V  + (A C - B ) E

                         2
 + %I SQRT(A E  - 2 %I B E - C)

          2                                                 2     2
     SQRT(B  - A C) ((%I A E - B) V + %I B E - C) SQRT(V  + E )
ASIN(------------------------------------------------------------))
                              2    2            2    2
                     (A C - B ) V  + (A C - B ) E

                   2                                   2
/(2 E SQRT(A E  - 2 %I B E - C) SQRT(A E  + 2 %I B E - C))

(C30) N1:(1-TSI)/2$

(C31) N2:(1+TSI)/2$

(C32) U: (N1*U1+N2*U2)*%E^(%I*(N1*A1+N2*A2))$

(C33) P: (N1*P1+N2*P2)*%E^(%I*(N1*B1+N2*B2))$

(C34) DUDTSI:DIFF(U,TSI)$

(C35) I1:N1*(K1*U+K2*DUDTSI)$

(C36) I1:EXPAND(I1)$

(C37) INT(X):=INTEGRATE(X,TSI)$

(C38) F1:MAP(INT,I1)$

(C39) EV(F1,TSI=1) - EV(F1,TSI=-1)$

(C40) ANSWER:RADCAN(%);
```

```
(D40) - ((((%I A2  + (- 2 %I A1 - 1) A2 + %I A1  + A1) %E       + %E       A2
                                                                    %I A1
 - A1 %E      ) K2 + ((2 A2 - 2 A1 + 4 %I) %E       + 2 %E       A2
                                                                    %I A1
 + (- 2 A1 - 4 %I) %E      ) K1) U2 + (((A2 - A1) %E       + %E       A2
                                                                    %I A1   2
 + (- 3 A1 - %I) %E       A2  + (3 A1  + 2 %I A1 - 1) %E       A2
                                                                    %I A1   2
 + (- A1  - %I A1  + A1) %E      ) K2 + (- 4 %I %E       - 2 %I %E       A2
                                                                    %I A1
 + (4 %I A1 - 4) %E       A2 + (- 2 %I A1  + 4 A1 + 4 %I) %E      ) K1) U1)
    3          2       2          3
/(A2  - 3 A1 A2  + 3 A1  A2 - A1 )
```

3.3 Matrix Manipulation

The MACSYMA System has provisions for all of the usual matrix manipulations including the computation of determinants, characteristic polynomials, and inverses. Matrix multiplication is performed by using the dot operator, "." (this operator can also be used for other non-commutative operations). Exponentiation is performed by using "^^". Thus, M.M is equivalent to M^^2 and the inverse of M, if it exists, is M^^(-1). The arithmetic operations (+), subtraction (-), multiplication (*), and division (/) are element-by-element operations. Illustrations of the use of selected matrix functions appear below.

```
(C1) batch("demo.max");

(C2) RATMX:TRUE$

(C3) MAT1:MATRIX([A,3+A,B],[2*B,A+B,5],[A-6,B-1,A*B]);
                                [   A    A + 3   B  ]
                                [                   ]
(D3)                            [ 2 B   B + A   5  ]
                                [                   ]
                                [ A - 6  B - 1  A B ]

(C4) DETERMINANT(MAT1);
         3       2                2     3     2                2
(D4)   2 B  + (- A  - 7 A + 4) B  + (A  - A  + A) B + 5 A  - 10 A - 90

(C5) MAT2:ENTERMATRIX(3,3);


Is the matrix  1. Diagonal  2. Symmetric  3. Antisymmetric  4. General
Answer 1, 2, 3 or 4
4;

Row 1 Column 1:  1;

Row 1 Column 2:  2;

Row 1 Column 3:  3;

Row 2 Column 1:  4;

Row 2 Column 2:  5;

Row 2 Column 3:  6;

Row 3 Column 1:  7;

Row 3 Column 2:  8;

Row 3 Column 3:  9;

Matrix entered.
                                [ 1  2  3 ]
                                [         ]
(D5)                            [ 4  5  6 ]
                                [         ]
                                [ 7  8  9 ]
```

```
(C6) CHARPOLY(MAT2,LAMBDA);
                                  3                 2
(D6)                     - LAMBDA   + 15 LAMBDA   + 18 LAMBDA

(C7) SOLVE(%);

                  3 SQRT(33) - 15              3 SQRT(33) + 15
(D7)  [LAMBDA = - ---------------, LAMBDA = ---------------, LAMBDA = 0]
                         2                            2

(C8) MAT1+MAT2;
                         [   A + 1      A + 5      B + 3  ]
                         [                                ]
(D8)                     [ 2 B + 4  B + A + 5       11    ]
                         [                                ]
                         [   A + 1      B + 7     A B + 9 ]

(C9) MAT1 - MAT2;
                         [   A - 1      A + 1      B - 3  ]
                         [                                ]
(D9)                     [ 2 B - 4  B + A - 5      - 1    ]
                         [                                ]
                         [  A - 13      B - 9     A B - 9 ]

(C10) MAT1*MAT2;
                         [      A       2 (A + 3)    3 B  ]
                         [                                ]
(D10)                    [     8 B      5 (B + A)    30   ]
                         [                                ]
                         [ 7 (A - 6)    8 (B - 1)   9 A B ]

(C11) MAT1.MAT2;


(D11)/R/

     [    7 B + 5 A + 12          8 B + 7 A + 15           9 B + 9 A + 18      ]
     [                                                                         ]
     [    6 B + 4 A + 35          9 B + 5 A + 40          12 B + 6 A + 45      ]
     [                                                                         ]
     [ (7 A + 4) B + A - 10  (8 A + 5) B + 2 A - 17  (9 A + 6) B + 3 A - 24    ]

(C12) MAT1/MAT2;
                              [          A + 3     B  ]
                              [   A      -----     -  ]
                              [            2       3  ]
                              [                       ]
                              [   B      B + A     5  ]
(D12)                         [   -      -----     -  ]
                              [   2        5       6  ]
                              [                       ]
                              [ A - 6    B - 1    A B ]
                              [ -----    -----    --- ]
                              [   7        8       9  ]
```

```
(C13) MAT3:MINOR(%,1,2);
                                        [   B     5  ]
                                        [   -     -  ]
                                        [   2     6  ]
(D13)                                   [            ]
                                        [ A - 6  A B ]
                                        [ -----  --- ]
                                        [   7     9  ]

(C14) MAT3^^(-1);
                     [        14 A B                       105            ]
                     [   ----------------     - ------------------------  ]
                     [         2                        2                 ]
                     [   7 A B  - 15 A + 90      7 A B  - 15 A + 90       ]
(D14)/R/             [                                                    ]
                     [       18 A - 108                 63 B              ]
                     [ - ------------------    ----------------------     ]
                     [          2                       2                 ]
                     [    7 A B  - 15 A + 90   7 A B  - 15 A + 90         ]

(C15) %.MAT3;
                                           [ 1  0 ]
(D15)/R/                                   [      ]
                                           [ 0  1 ]

(C16) TRANSPOSE(MAT3);
                                         [ B  A - 6 ]
                                         [ -  ----- ]
                                         [ 2    7   ]
(D16)                                    [          ]
                                         [ 5   A B  ]
                                         [ -   ---  ]
                                         [ 6    9   ]

(C17) MAT2[2,2]:NEW$

(C18) MAT2;
                                        [ 1   2   3 ]
                                        [           ]
(D18)                                   [ 4  NEW  6 ]
                                        [           ]
                                        [ 7   8   9 ]

(C19) I:ROW(MAT2,3);

(D19)                                    [ 7  8  9 ]
```

```
(C20) MAT4:ADDROW(MAT1,I);

                                [   A     A + 3    B   ]
                                [                      ]
                                [ 2 B     B + A    5   ]
(D20)                           [                      ]
                                [ A - 6   B - 1   A B  ]
                                [                      ]
                                [   7       8      9   ]

(C21) IDENT(4);
                                [ 1  0  0  0 ]
                                [            ]
                                [ 0  1  0  0 ]
(D21)                           [            ]
                                [ 0  0  1  0 ]
                                [            ]
                                [ 0  0  0  1 ]

(C22) ECHELON(MAT1);
                          [     A + 3          B          ]
                          [ 1   -----          -          ]
                          [       A          A          ]
                          [                             ]
                          [                    2        ]
(D22)/R/                  [                 2 B  - 5 A  ]
                          [ 0     1         ------------ ]
                          [                          2  ]
                          [                 (A + 6) B - A ]
                          [                             ]
                          [ 0     0         1            ]

(C23) TRIANGULARIZE(MAT4);
          [ 7       8                      9                      ]
          [                                                       ]
          [ 0  - 9 B + 7 A                - 18 B + 35             ]
(D23)/R/  [                                                       ]
          [                    2                          2      ]
          [ 0       0      - 9 B  + (16 A + 54) B - 9 A  + 5 A - 105 ]
          [                                                       ]
          [ 0       0                      0                      ]

(C24) H[I,J]:=1/(I+J-1)$

(C25) GENMATRIX(H,5,5,2,2);

                                [ 1  1  1  1 ]
                                [ -  -  -  - ]
                                [ 3  4  5  6 ]
                                [            ]
                                [ 1  1  1  1 ]
                                [ -  -  -  - ]
                                [ 4  5  6  7 ]
(D25)                           [            ]
                                [ 1  1  1  1 ]
                                [ -  -  -  - ]
                                [ 5  6  7  8 ]
                                [            ]
                                [ 1  1  1  1 ]
                                [ -  -  -  - ]
                                [ 6  7  8  9 ]

(D26)                           BATCH DONE
```

3.4 Generation of FORTRAN Code

MACSYMA also has the capability to produce FORTRAN code. This feature is extremely useful in cases where you may wish to incorporate your MACSYMA results into existing FORTRAN programs or use the results to create new programs on another computer.

MACSYMA expressions can be converted to FORTRAN by typing

```
FORTRAN (expression 1);

        .
        .
        .

FORTRAN (expression n):
```

One method for transferring FORTRAN generated code to other computers is to use an intelligent terminal equipped with a cassette tape (or microcomputer and diskette) to store the information. Later the stored information can be read into another computer. The HP 2647A terminal is useful for this purpose in that the programmable keys can help you to efficiently remove labels from your expressions before they are saved on tape. Alternately, WRITEFILE and CLOSEFILE could be used to create a file of the portion of your MACSYMA session containing the FORTRAN expressions. An editor could then be used to edit out labels and other extraneous information. Illustrations of the use of the FORTRAN command appear below.

```
(C1) a:matrix([1,0],[0,c]);
                                  [ 1  0 ]
(D1)                              [      ]
                                  [ 0  C ]

(C2) b:matrix([1,1],[c,0]);
                                  [ 1  1 ]
(D2)                              [      ]
                                  [ C  0 ]

(C3) p:(x-y)*(x^2+x*y+y^2);
                                    2          2
(D3)                      (X - Y) (Y  + X Y + X )

(C4) q:x-3*a/(x^2+4*b);
                          [       3             ]
                          [ X - ------      X    ]
                          [       2             ]
                          [     X  + 4          ]
(D4)                      [                     ]
                          [                 3 C ]
                          [     X       X - --- ]
                          [                 2   ]
                          [                X    ]

(C5) fortran(a);


      A(1,1) = 1
      A(1,2) = 0
      A(2,1) = 0
      A(2,2) = C
(D5)                                DONE

(C6) fortran(p);
      (X-Y)*(Y**2+X*Y+X**2)
(D6)                                DONE

(C7) r:s=p;
                                      2          2
(D7)                    S = (X - Y) (Y  + X Y + X )

(C8) fortran(r);
      S = (X-Y)*(Y**2+X*Y+X**2)
(D8)                                DONE

(C9) fortran(q);
      Q(1,1) = X-3/(X**2+4)
```

```
      Q(1,2) = X
      Q(2,1) = X
      Q(2,2) = X-3*C/X**2
(D9)                                  DONE



(C11) answer;
                      2
(D11) - (SQRT(A E  + 2 %I B E - C)

                                                                  2    2
       SQRT(A C - B ) ((%I A E + B) V + %I B E + C) SQRT(V  + E )
 ASINH(-------------------------------------------------------------)
                                2  2          2  2
                     (A C - B ) V  + (A C - B ) E

                      2
 + SQRT(A E  - 2 %I B E - C)

                                                                  2    2
       SQRT(A C - B ) ((%I A E - B) V + %I B E - C) SQRT(V  + E )
 ASINH(-------------------------------------------------------------))
                                2  2          2  2
                     (A C - B ) V  + (A C - B ) E

                  2                                2
/(2 E SQRT(A E  - 2 %I B E - C) SQRT(A E  + 2 %I B E - C))

(C12) fortran(answer);


      -(SQRT(A*E**2+2*(0.0,1.0)*B*E-C)*ASINH(SQRT(A*C-B**2)*(((0.0,1.0)*
     1   A*E+B)*V+(0.0,1.0)*B*E+C)*SQRT(V**2+E**2)/((A*C-B**2)*V**2+(A*C
     2   -B**2)*E**2))+SQRT(A*E**2-2*(0.0,1.0)*B*E-C)*ASINH(SQRT(A*C-B**
     3   2)*(((0.0,1.0)*A*E-B)*V+(0.0,1.0)*B*E-C)*SQRT(V**2+E**2)/((A*C-
     4   B**2)*V**2+(A*C-B**2)*E**2)))/(E*SQRT(A*E**2-2*(0.0,1.0)*B*E-C)
     5   *SQRT(A*E**2+2*(0.0,1.0)*B*E-C))/2.0
(D12)                                 DONE

(C13)
```

3.5 MACSYMA Editor

MACSYMA has a very useful editor which can be used to correct syntax errors, correct typing errors, or modify previous instructions. To correct syntax errors in the current MACSYMA command string type <ESC> <CR>. The system will indicate that you are in the MACSYMA editor and will redisplay the command string. You can edit the string by using any of the commands listed in Table 3.2.

Table 3-2

Some MACSYMA Editing Commands

Command	Function
B	Moves the cursor to end of input string
C (or nC)	Moves the cursor one character (or n characters) to the right
-C (or -nC)	Moves the cursor one character (or n characters) to the left
D (or nD)	Delete one character (or n characters) to the right of the cursor
-D (or -nD)	Delete one character (or n characters) to the left of the cursor
I<string>	Insert string to the left of the cursor
J	Move the cursor to the beginning of input string
P	Reprint the input string
S<string> (or nS<string>)	Move the cursor to the right of the occurrence (or nth occurrence) of string
-S<string> (or -nS<string>)	Move the cursor to the left of the occurrence (or nth occurrence) of string

An expression may be edited by using a single command or multiple commands. When the insert or search command is used in conjunction with other commands, use an escape <ESC> to separate them. Two consecutive escapes followed by return cause the modified expression to be redisplayed. If no further editing is required, two additional escapes followed by carriage return will return the expression to MACSYMA.

In order to modify a previous MACSYMA command string, use the string function. For example, to edit command-line C8 when the current command-line is C15, type

```
(C15) STRING(C8);
```

The string function will redisplay the command-line. To enter the MACSYMA editor, type <ESC> <CR> and edit the input string as indicated above.

A sample illustration of the use of the MACSYMA editor appears below. Explanatory comments are delimited by (* and *) and appear in the right column. The symbol "$" represents the escape or altmode key. It also denotes the cursor for the MACSYMA editor.

MACSYMA EDITOR

COMMENTS

```
(C10) 23*sqrt(y^5 + 8y - 3)/100;
"Y" is not an infix operator.
23 * SQRT ( Y ^ 5 + 8 Y **$** - 3 ) / 100
Please rephrase or edit.

(C10) $ <CR>                                        (* enter the editor *)
In editor:  (Type just <Alt><Alt> to exit.)
$23*sqrt(y^5 + 8y - 3)/100
$s8$i*$$ <CR>                                       (* locate "8" and insert * following
23*sqrt(y^5 + 8*$y - 3)/100                            it *)
$$ <CR>                                             (* return to MACSYMA *)
(C10) 23*sqrt(y^5 + 8*y - 3)/100;

                                 5
                          23 SQRT(Y  + 8 Y - 3)
(D10)                     ---------------------
                                  100

(C11) string(c2);                                   (* get command C2 for editing *)
(D11)                     F(X):=(X^2-3*X+5)/X^3

(C12) $ <CR>                                        (* enter the editor *)
In editor:
$F(X):=(X^2-3*X+5)/X^3
$9c <CR>                                            (* move 9 characters to the right,
F(X):=(X^8$-3*X+5)/X^3                                 delete the next character (2),
$$ <CR>                                                and insert "8" *)
(C12) F(X):=(X^8-3*X+5)/X^3;                        (* return to MACSYMA *)

                                   8
                                  X  - 3 X + 5
(D12)                  F(X) := ------------
                                     3
                                    X

(C13) 5/(x-4)^3;

                                  5
(D13)                         --------
                                     3
                              (X - 4)

                                                    (* get last command string for
(C14) $ <CR>                                           editing *)

In editor:
$5/(x-4)^3
$d;100$$ <CR>                                       (* delete first character (5) and
100$/(x-4)^3                                           insert "100" *)
$$ <CR>                                             (* return to MACSYMA *)
(C14) 100/(x-4)^3;

                                 100
(D14)                         --------
                                     3
                              (X - 4)
```

3.6 MACSYMA File Commands

MACSYMA has a variety of batch and storage functions for manipulating files of MACSYMA commands or expressions. The BATCH functions are useful for manipulating files that may be created using a text editor, while the storage functions permit the user to specify that certain MACSYMA expressions should be written to, or retrieved from disk. A summary of the more frequently used commands appears in Table 3-3, and a sample MACSYMA interaction follows the table of commands.

TABLE 3-3

MACSYMA FILE COMMANDS

Commands	Descriptions
batch ("file-specification")	Reads and evaluates MACSYMA command lines from a file.
demo("file-specification")	Similar to BATCH except MACSYMA pauses after each command line and continues processing when <CR> is typed. Typing any other character followed by <CR> terminates the demo.
batchload("file-specification")	Identical to BATCH except no terminal output is generated.
batcon (batcount + 1)	Resumes reading a previously referenced BATCH file, beginning with the statement following the last expression read.
save("file-specification",	Writes selected portions expressions of a MACSYMA session to a file.
loadfile("file-specification")	Loads a file of quantities saved from a prior MACSYMA session.
writefile ("file-specification")	Opens a file for writing and records all interactions between the user and MACSYMA.
closefile ("file-specification")	Closes a file opened by WRITEFILE.

Note: When your current working directory contains the file to be manipulated, the "file-specification" need only consist of the file name. Otherwise, the full pathname for the file is required.

Demonstration of MACSYMA File Commands

```
(c1)  batch("mactest.max");
/*    MACSYMA TEST          */
(c2)  PRINT "THIS IS A TESTFILE"$

"THIS IS A TEST FILE" is not an infix operator.
print THIS IS A TEST FILE **$**
Please rephrase or edit.

Error in batch file
(c2) $ <CR>
In editor:
$PRINT"THIS IS A TEST FILE"
ST$I($BI)$$ <CR>

PRINT("THIS IS A TEST FILE")$

THIS IS A TEST FILE

(c3)  BATCON(BATCOUNT+1);
(c4)  PRINT("NOW I HAVE LEARNED THE ERROR OF MY WAYS")$

NOW I HAVE LEARNED THE ERROR OF MY WAYS

(c5)  demo("test1.max");

(c6)  EXP:X^2-3*X*Y+2*Y^2;

(d6)                                        2 y^2 - 3 x y + x^2

(c7)  EXP2:FACTOR(EXP);

(d7)                                        (y - x)  (2 y - x)

(d8)                                  BATCH DONE

(c9)  save("test2.max",exp2);

(d9)                              [test2.max, exp2]

(d10)                                 BATCH DONE

.
.
.

% MACSYMA
This is UNIX MACSYMA Release 309.2
(c) 1976, 1983 Massachusetts Institute of Technology.
All Rights Reserved.
(c1)  loadfile("test2.max"):
test2.max being loaded.
```

```
(d1)                                    done

(c2)  EXP2;
(d2)
                                        (y - x)  (2 y - x)
```

4.0 FILES AND DIRECTORIES

A file is an area on a particular storage device which contains data or text. UNIX files are organized in directories which catalog the files associated with particular users. Each user has a working directory that he is assigned to when he logs in. You can determine the pathname of your working directory by typing

```
pwd  <CR>
```

UNIX files are identified by a base name and an optional extension which is separated from the base name by ".". File names usually contain up to fourteen characters and do not include the character "/". File extensions are used to describe the type of information stored in files, and the system has several defaults for special purposes. For example, ".c" and ".f" are used for C and FORTRAN source files, respectively.

5.0 UNIX COMMANDS

At login, you are assigned to the UNIX C shell[2]. The prompt character is the percent (%) sign, and it indicates that the machine is waiting for you to enter a command to perform some action. UNIX commands should be typed in lower case and are terminated with carriage return. A command, along with options, an expression, and arguments, can contain a maximum of 256 characters. Numerous commands are available to communicate with the UNIX Operating System. Some of the more commonly used commands are introduced below.

[2]For information on changing shell scripts, consult the UNIX Reference Manual.

5.1 Commands to Manipulate Files/Directories

There are a variety of UNIX commands that can be used to manipulate files and directories. Some of the more useful commands appear in Table 5-1.

Table 5-1

Directory and File Commands

Command	Function
ls	List the files stored in a directory.
cat	Displays the contents of a file.
nl	Displays the contents of a file with line numbers.
more	Displays the contents of a file and pauses when the screen fills with information. Pressing the spacebar prints another screen, return prints another line, and q terminates the display.
head (-n)	Prints out the first few lines of a file as specified by n. The default is 10 lines.
tail (-n)	Prints out the last few lines of a file as specified by n.
cd	Changes you to your working directory.
cd pathname	Moves you to the directory specified by pathname.
mv file 1 file 2	Changes the name of file 1 to file 2. The location can be changed also if file 2 specifies a full pathname.
cp file 1 file 2	Makes file 2 a duplicate copy of file 1.
rm file(s)	Eliminates a specific file or files.
pr	Formats a file for printing on the line printer. The output comes to the standard output file (terminal) unless otherwise directed.
cmp file 1 file 2	Compares file 1 and file 2 and indicates the first character and line where any difference occurs.
diff file 1 file 2	Indicates what lines differ in the two files.

5.2 Commands to Obtain Information

The UNIX operating system is user friendly in that it provides information regarding the syntax and use of certain commands interactively. It also provides interactive information regarding the system environment. A brief summary of some of these commands is presented in Table 5-2.

Table 5-2

Commands to Obtain Information

Command	Function
man	Provides manual information on a specified keyword.
apropos	Lists the manual sections containing the specified keyword.
learn	Gives CAI courses and practice in the use of some features of UNIX
stty	Lists terminal characteristics.
date	Gives current date and time.
finger	Lists information on all users currently on the system.
users	Generates a compact list of all users on the system.
printenv	Prints out values of variables in the environment.
set	Shows values of all variables currently defined in the shell.

Illustrations of the output generated by the commands in Table 5-2 appear below.

```
% stty
new tty, speed 1200 Baud; -tabs ff1
```

```
% date
Mon June 30 13:45:29 EDT 1986
```

```
% finger
Login            Name                TTY Idle          When          Office
mam          Magaline Macklin          09          Mon 13:41
sjd          Sandra J. DeLoatch        10          Mon 13:26          x2-2062
```

```
% users
mam sjd
```

```
% printenv
HOME=/usr/nasa/mam
SHELL=/bin/csh
PATH=:/usr/ucb:/bin:usr/bin
TERM=vt100
USER=mam
```

```
% set
arav     ( )
cwd      /usr/nasa/mam
home     /usr/nasa/mam
path     (./usr/ucb /bin /usr/bin)
prompt   %
shell    /bin/csh
status   0
term     vt100
user     mam
```

5.3 Quit Commands and Control Characters

There are several control characters which have a special meaning to the UNIX operating system. Frequently used characters and their functions are:

^U (kill)	Cancels a line
^S (stop)	Suspends terminal output
^Q (start)	Resumes terminal output
^C (intr)	Interrupts the execution of a job or process
^Z (susp)	Suspends a running process or command
^D (eof)	Denotes end of file

The stty command can be used to change the meaning of any of these characters.

5.4 Electronic Communication Commands

A useful feature of the UNIX computing environment is that it permits interactive communication among users. The interactive communication facilities permit users to send one-way messages or to communicate directly by simulating a telephone call. The communication commands are described below.

mail	Used to send or receive messages
write	Allows direct communication with other users (one-way communication)
talk	Used to communicate directly with other users by simulating a two-way conversation.
mesg n	Prevents anyone from writing or talking to you.
mesg y	Reverses the effect of mesg N.

5.4.1 Mail

The mail command permits you to receive messages or to send messages to other users on system. When you have a message from another user, the system notifies you by printing

```
You have mail
```

when you log in.

In order to read your messages, type

```
mail <CR>
```

Mail will respond by typing its version number, the date, and a list of the messages you have waiting. Next, in response to the mail prompt, & enter.

```
t 1 <CR>
```

And your first message will be displayed. If you have several new mail messages, continue entering T and the appropriate message number. You can delete messages by typing D after reading the message. Messages which are not deleted are saved automatically in the file mbox in your login directory.

To send a message to another user, type

```
mail login name  <CR>

message text

  .
  .
  .
^D
```

5.4.2 Write

The write command permits you to communicate directly with other users on the system. To initiate communication with another user, type

```
write username ttyname

message

.
.
.
```

For example,

```
write mac ttu08

I am ready.
```

Would cause the following information to be displayed on the terminal of user mac:

```
Message from nirvana!  Mac on tty08 at . . .

I am ready.
```

User mac should write back at this point. Communication between two users continues until an interrupt is sent.

When only one individual is logged on with a given user name, ttyname can be omitted. The finger command can be used to obtain information concerning the users currently on the system.

5.4.3 Talk

Talk permits you to communicate directly with another user by simulating a conversation. The command can be used successfully on video display terminals with direct cursor positioning. In order to talk to another user, enter

```
talk user name [ttyname] <CR>
```

The following message will be sent to the terminal of the person you wish to communicate with:

```
message from talk-daemon   nirvana at . . .

talk:  connection requested by user name   nirvana

talk:  respond with:  talk user name   nirvana
```

Once the recipient has responded, each person's terminal screen is split into two windows. One window displays your message and the other displays the recipient's response. The conversation can be terminated by typing an interrupt character.

6.0 EDITING FILES

There are several editors available under UNIX for creating or modifying files. One of these is a line-oriented editor called ed. In order to create a file using ed, type ed

ed will respond by printing

At this point, since the input file does not exist, text may be inserted in the buffer using an insert (i) or append (a) command. The contents of the buffer can then be saved or modified by issuing certain other commands. Line editing commands allow users to manipulate lines of text that are inserted into the text buffer. In order to insert text into an empty buffer, type

```
i <CR>
first line of text <CR>
     .
     .
     .
last line of text <CR>
```

"." terminates the insertion. The inserted text can now be manipulated by specifying a line number or range of lines and a command. Line numbers are automatically assigned by ed when text is entered into the buffer. When no range or line number is specified with a command, the current line is manipulated. To determine which line is current, type

```
. <CR>
```

For example, to change all occurrences of "xyz" to "abc" in line 4 of the text buffer, the following command is required

```
(4S)/xyz/abc/  <CR>
```

If line 4 is the current line, then

```
s/xyz/abc/
```

will suffice. When your line editing session is complete, type

```
w   <CR>
q   <CR>
```

to terminate the session and save the results of your edits. If the file name is omitted, the editing results are saved under the file name specified when ed was invoked. The q command should be used alone when you do not wish to save the contents of your editing session.

Numerous line editing commands are available to manipulate text. Some of the more frequently used commands and ranges of text are described in table 6-1.

Table 6-1

Line Editing Commands

Functions	Commands
Replacing Lines	
Replace m through n with k lines	m,nc first line of text . . . line k of text .
Replace line 5 with single line of text	5c new line .
Printing Lines/Locating Text	
Print current line	.
Print current line number	.=
Print lines m through n	m,np
Print contents of the buffer	1,$p
Print all lines that contain a given string	g/string/p
Locate the first occurrence of a given string and print the line number where it occurs	/string/=
Substituting Text	
Replace next occurrence of a given string in the current line	s/old-string/new-string/
Copying and Moving Text	
Copy same text in lines m through n to the end of buffer	m,nt$
Delete text in lines 40 through 50 and insert it before line 80	40,50m80

7.0 USER AIDS FOR MACSYMA

Numerous resources and facilities are available to provide assistance with the use of MACSYMA. Written resources include

MACSYMA Reference Manual,
MACSYMA Primer
MACSYMA Newsletter
MACSYMA Applications Newsletter

The reference manual completely describe the MACSYMA system. It lists all the commands, functions, switches, and options that are available for use. The MACSYMA Primer provides a brief introduction to MACSYMA which includes examples and illustrations of MACSYMA syntax. Both newsletters will provide general information of interest to the computer algebra community as well as applications of MACSYMA to real world problems. The first three documents are available from Symbolics, Inc. (11 Cambridge Center, Cambridge, MA 02142), while the last document may be obtained from Paradigm Associates, Inc. (29 Putman Avenue, Suite 6, Cambridge, MA 02139). Reference copies of the written resources are available in CAB. In addition to the written resources, there are interactive aids that users may take advantage of on-line. These aids are described below.

7.1 Demonstration and Share Files

The DEMO directory contains files that provide interactive demonstrations of how MACSYMA functions work on real problems. To obtain a listing of the files available, type

```
(cd /usr/macsyma.309/demo <CR>)
ls <CR>
```

In order to run a DEMO file, load MACSYMA, and after the command line number appears, type

```
batch ("/usr/macsyma.309/demo/filename"); <CR>
  or
demo ("/usr/macsyma.309/demo/filename"; <CR>
```

The batch command generates a listing of the entire file with no pauses between command lines. The demo command produces a pause after each command display. To execute the next command, type <CR>.

The SHARE directory provides application programs contributed by members of the MACSYMA user community. A listing of the available programs may be obtained by typing

```
cd /usr/macsyma.309/share <CR>
ls <CR>
```

share files of type "USE" provide the documentation for the use of programs and should be queued to the line printer. Files of type "DEM" give demonstrations of the programs. To use one of the share files, type

```
batch("usr/macsyma.309/filename.mac"); <CR>
  or
loadfile("/usr/macsyma.309/filename. l  ") <CR>
```

That is, use batch for MACSYMA (mac) files and loadfile for LISP (l) files.

7.2 Help Commands

MACSYMA has some very useful commands to enable the user to obtain information on-line of particular benefit to new users is

primer (help); <CR>

This command provides an interactive version of the MACSYMA Primer. Other useful commands are described below.

Command	Function
apropos (string)	Prints a listing of all MACSYMA commands or switches that contain "string"
describe (command)	Displays the description of "command" that appears in the MACSYMA Reference Manual
example (command)	Provides examples of the use of certain commands on selected functions or expressions in demo mode

Apropos is extremely useful when the exact name for a command does not come to mind, but a string of characters that the name contains does. On the other hand, describe and example are helpful when a command name poses no problem, and a quick description or illustration of its use is required.

7.3 MACSYMA Break

Suppose you have entered the following command:

```
for n:1 thru 100 do f[n]: ratsimp(g[n])$
```

If the command has not terminated after a few minutes, you may wonder if the system is still working on the first expression, or is nearing the 100th expression. The calculations can be interrupted in order to explore the status of the computation by typing

The machine's response is

```
interrupt (h for help):
```

At this point, you may type

```
m; <CR>
```

to enter a MACSYMA break. Next, type

```
n; <CR>
```

to determine the value of N. If the system indicates that N is still 1, you may wish to cancel the command and change your expression.

```
r; <CR>
```

to cancel the command and return to MACSYMA. To exit the MACSYMA break and continue the computation, type

```
exit; <CR>
```

7.4 Storage Management Commands

Computer algebra systems use large amounts of memory, and one of the major problems you might encounter is running out of space. For this reason, you should periodically remove large expressions, function definitions, or input and output lines from your allocated space, if they are no longer needed. The MACSYMA KILL command can be used for this purpose. The command can have any one of the following forms:

```
kill (expression) $
kill (c-labels) $
kill (d-labels) $
kill (labels) $
kill (all) $
kill (expression1, expression2, ...) $
```

When a large amount of memory has been used, you will frequently get a fatal error from the system and will not be able to recover your work. When your problem requires a large amount of memory, you might want to experiment with the alloc command. For example, try typing

```
alloc(LIST,1000); <CR>
```

at the beginning of your MACSYMA session to obtain additional list space. Other space parameters that can be manipulated are FIXNUM, FLONUM, BIGNUM, SYMBOL, and ARRAY.

Another method for managing storage problems is to save the significant portions of your MACSYMA session for further development in a new MACSYMA session. This can be accomplished by typing

```
save("file-specification",list of items); <CR>
```

The list of items may contain any one of the following:

values	saves all expressions which are not labels or arrays
labels	saves all labeled expressions
expression(s)	saves any named expression(s)
arrays	saves all declared and undeclared arrays
functions	saves all user-defined functions

The stored expressions can be loaded into in a new MACSYMA session, by using the loadfile command.

APPENDIX A

UNIX COMMAND INDEX

COMMAND	FUNCTION
apropos	lists the manual sections containing the specified keyword
cat	displays the contents of a file
cd pathname	changes the current directory to the one specified by pathname
cmp f1 f2	compares f1 and f2 and indicates the first character and line where any difference occurs.
cp f1 f2	make file f2 a duplicate copy of f1
date	prints the current date and time
diff f1 f2	indicates what lines differ in files f1 and f2
finger	lists information on all users currently on the system
head (-n)	prints out the first few lines of a file as specified by n
learn	provides CAI courses and practice in the use of some of the features of UNIX
ls	lists the files stored in a directory
man (keyword)	provides manual information on a specified keyword
more	displays the contents of a file and pauses when the screen fills with information. Pressing the spacebar prints another screen, returns prints another line, and q terminates the display.
nl	displays the contents of a file with line numbers
pr	formats a file for printing on the line printer. The output cones to the standard output file (terminal) unless otherwise directed.
printenv	prints out values of variables in the environment
set	shows values of all variables currently defined in the shell

stty	lists terminal characteristics
tail (-n)	prints out the last few lines of a file as specified by n
users	generates a compact list of all users on the system.

APPENDIX B

MACSYMA CONTROL CHARACTERS

^Z, ^D	Enters and exits LISP from MACSYMA.
^L, ^K	Line feed.
^[,$	Enters MACSYMA editor.
^[^[,$$	Exits MACSYMA editor.
^I	Tabs.
^B	Enters LISP.
^C	Enters MACSYMA interrupt. Interrupt options are:

```
l,      Enters LISP.
m,      Enters MACSYMA break.
h,      Prints interrupt options.
e,      Returns from interrupt.
r,      Reset, return to top level.
q,      Quit MACSYMA
exit;   Returns to top level MACSYMA.
```

BIBLIOGRAPHY

Wang, Paul S.: Symposium on Symbolic and Algebraic Computation. Association for Computing Machinery, 1981.

DeLoatch, Sandra J.: MACSYMA User's Guide. NASA CR-166059, 1983.

Golden, V. Ellen: An Introduction to ITS for the MACSYMA User. Laboratory for Computer Science, Massachusetts Institute of Technology, 1981.

Golden, V. Ellen: MACSYMA Primer. Laboratory for Computer Science, Massachusetts Institute of Technology, 1982.

Golden, V. Ellen, and McGin, Mickey: Proceedings of the 1984 MACSYMA User's Conference.

Joy, W. N., et al: UNIX Programmer's Manual. University of California, Berkley, 1981.

Lewis, V. Ellen: Proceedings of the 1979 MACSYMA User's Conference. Laboratory for Computer Science, Massachusetts Institute of Technology, 1979.

Golden, V. Ellen, and McGinn, Mickey: Proceedings of the 1984 MACSYMA User's Conference.

MATHLAB Group: MACSYMA Reference Manual. Laboratory for Computer Science, Massachusetts Institute of Technology, 1978.

Moses, Joel: Algebraic Simplification, A Guide for the Perplexed. Communications of the ACM, Vol. 14, 1971, pp. 548-560.

Moses, Joel: Symbolic Integration. Massachusetts Institute of Technology, 1967.

Meals, Kenton: Use of MACSYMA in the Navy Laboratory Research Community. D. W. Taylor Naval Ship Research and Development Center, 1979.

Pavelle, Rothstein, and Fitch: Computer Algebra. Scientific American, Vol. 245, 1981, pp. 136-153.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-172518 (Revision 1)		

4. Title and Subtitle	5. Report Date
MACSYMA Usage at Langley	January 1987
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
Sandra Jean DeLoatch	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Norfolk State University Foundation 2401 Corprew Avenue Norfolk, VA 23504	NAS1-16921
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Contractor Report
National Aeronautics and Space Administration Washington, DC 20546	14. Sponsoring Agency Code
	505-90-21-02

15. Supplementary Notes

Langley Technical Monitor: John N. Shoosmith
(Supersedes NASA CR-172518, dated February 1985)

16. Abstract

MACSYMA, a large and versatile symbolic manipulation system, is discussed in relation to its access and use at NASA, Langley Research Center. The report provides introductory information for new users, descriptions of the available resources for new and experienced users, and illustrations of selected features of the MACSYMA computing environment.

17. Key Words (Suggested by Author(s)) (STAR category underlined)	18. Distribution Statement
Computer algebra MACSYMA NASA, LaRC Symbolic manipulation	Unclassified - Unlimited Subject Category 59

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	57	A04

For sale by the National Technical Information Service, Springfield, Virginia 22161